

MO417 — Complexidade de Algoritmos I — 2s2019

Lista de Exercício 4

Além dos exercícios abaixo, recomendo que façam a maior quantidade possível de exercícios dos livros texto (CLRS e Manber) dos capítulos relacionados.

Questão 1. (Manber) Sejam x_1, x_2, \dots, x_n uma sequência de números reais (não necessariamente positivos). Projete um algoritmo $O(n)$ para encontrar a subsequência x_i, x_{i+1}, \dots, x_j (de elementos consecutivos) de tal forma que o produto dos números é máximo entre todas as subsequências consecutivas. O produto de uma subsequência vazia é definido como 1.

Questão 2. (Baseado em Solved Exercise 1 de Kleinberg e Tardos) Suponha que você tenha um vetor A de n números naturais. Suponha que esse vetor tem a seguinte propriedade:

- se o máximo ocorre em $A[p]$ para um índice p , então $A[1] \leq A[2] \leq \dots \leq A[p]$ e $A[p] \geq A[p+1] \geq \dots \geq A[n]$;
- se $A[i] = A[j] = x$, então $A[k] = x$ sempre que $i \leq k \leq j$.

Seu objetivo é encontrar o valor máximo nesse vetor. Escreva o algoritmo mais eficiente que conseguir. Justifique a complexidade e a correção.

Questão 3. (Manber) O quebra-cabeça das torres de Hanoi é um exemplo de um problema não trivial que tem uma solução simples recursiva. Há n discos colocado em uma estaca em ordem decrescente de tamanho. Há duas estacas livres. O objetivo do quebra-cabeças é mover todos os discos, um por vez, da primeira estaca até outra estaca da seguinte maneira. Discos são movidos do topo de uma estaca para o topo de outra. Um disco só pode ser movido se for menor do que todos os outros discos na estaca de destino. Em outras palavras, a ordenação dos discos em ordem decrescente deve ser mantida em todos os momentos. O objetivo é mover todos os discos com o menor número de movimentos.

- Projete um algoritmo (por indução) para encontrar uma sequência de movimentos que resolve as torres de Hanoi para o problema com n discos.
- Quantos movimentos são realizados pelo seu algoritmo? Construa uma relação de recorrência para o número de movimentos e resolva-a (exatamente).
- Mostre que o número de movimentos da parte (b) é ótimo, i.e., mostre que não pode existir outro algoritmo que realiza menos movimentos.

Questão 4. Considere o problema da subsequência consecutiva par máxima em que, dado um vetor, queremos encontrar uma subsequência (de números consecutivos) cuja soma seja um número par com o maior valor. Projete um algoritmo usando indução para esse problema. Analise o seu tempo de execução usando uma recorrência.

Questão 5. Considere o problema de encontrar o máximo em um vetor de n números. Projete um algoritmo de divisão e conquista para o problema. Experimente duas abordagens: usar um subproblema de tamanho $n - 1$ e dividir o subproblema em dois subproblemas de tamanho aproximadamente $n/2$. Qual a melhor abordagem?

Questão 6. (Kleinberg e Tardos, Solved Exercise 2) Suponha que você está prestando consultoria em uma empresa de investimentos e gostaria de saber em que dia se deve comprar e em que dia se deve vendê-la para maximizar o lucro. Suponha que você tenha a estimativa dos preços de n dias. Escreva um algoritmo que execute em tempo $O(n \log n)$ baseando-se no princípio da divisão e conquista.

Questão 7. Leia a seção sobre o algoritmo de Strassen de CLRS (seção 4.2 na 3ª Edição e seção 28.2 na 2ª Edição).